

1. Tempo disponibile 120 minuti (90 minuti per gli studenti di “Introduzione agli Algoritmi” - 6 CFU, che devono fare solo i primi 3 esercizi).
2. Non è possibile consultare appunti, slide, libri, persone, siti web, ecc.
3. Scrivere in modo leggibile, su ogni foglio, nome, cognome e numero di matricola.
4. Le soluzioni agli esercizi che richiedono di progettare un algoritmo devono:
 - spiegare a parole l'algoritmo (se utile, anche con l'aiuto di esempi o disegni),
 - fornire e commentare lo pseudo-codice (indicando il significato delle variabili),
 - calcolare la complessità (con tutti i passaggi matematici necessari),
 - se l'esercizio ammette più soluzioni, a soluzioni computazionalmente più efficienti e/o concettualmente più semplici sono assegnati punteggi maggiori.

1. Calcolare la complessità $T(n)$ del seguente algoritmo MYSTERY:

Algorithm 1: MYSTERY(INT n) \rightarrow INT

```
if  $n \leq 1$  then
  | return 1
else
  | INT  $v \leftarrow 0$ 
  | for  $i \leftarrow n$  downto 1 do
  |   |  $v \leftarrow v + i$ 
  |   end
  | return  $v + \text{MYSTERY2}(n)$ 
end

function MYSTERY2(INT  $m$ )  $\rightarrow$  INT
if  $m \leq 1$  then
  | return 1
else
  | INT  $w \leftarrow 0$ 
  | for  $j \leftarrow 1$  to  $m$  do
  |   |  $w \leftarrow w + j$ 
  |   end
  | return  $w + \text{MYSTERY}(m - 1)$ 
end
```

2. Dato un albero AVL inizialmente vuoto effettuare le seguenti operazioni in ordine e mostrare lo stato dell'albero dopo ogni operazione:

(a) insert 4 (b) insert 2 (c) insert 1 (d) insert 6 (e) insert 3
(f) insert 5 (g) insert 8 (h) insert 7 (i) delete 3 (j) delete 5

Indicare chiaramente quale operazione viene eseguita.

3. Progettare un algoritmo che dato un vettore $V[1..n]$ contenente n numeri tutti differenti fra di loro, ed un intero K , tale che $0 \leq K \leq n - 1$, restituisce quel valore x presente in V tale che V contiene esattamente K valori più piccoli di x e $(n - K - 1)$ valori più grandi di x . Discutere il costo dell'algoritmo nel caso ottimo, nel caso pessimo, e possibilmente anche nel caso medio.

4. Progettare un algoritmo che, dato un grafo non orientato connesso $G = (V, E)$ e due vertici $v_1, v_2 \in V$, stampa un cammino di lunghezza minima da v_1 a v_2 .